"QUICK" MODE

Quick (LTF)

FOR ZX81/TS1000/TS1500

Published

by Fred Nachbaur

We leave the wonderful world of true hi-res for a moment, to show you just one of any number of fascinating possibilities that can result from experimenting with the ZX81's interrupt-driven display routines.

Those of you who have played with Wilf Rigter's "Nova" package already know of some of the neat things you can do by "messing with the interrupts." For instance, smaller displays can be created to increase the time available for computing, thereby speeding up the machine by up to 4 times the normal "SLOW" mode rate.

We're all used to the FAST vs. SLOW dilemma. In FAST mode, the machine cruises right along, but you can't see the display. In SLOW mode you have a continuous display, but the overall speed often leaves much to be desired. Well, how about a FULL-SIZE, continuous display, while running almost three times as fast as the normal SLOW mode? You got it! The QUICK-mode routine in this article works with most BASIC or machine-code programs. The exceptions are other programs that already modify the interrupts for custom displays (as WRX16-based software, the Software Farm programs, possibly others).

Nothing comes for free, however. QUICK mode is a bit of a compromise, in that the display is slightly "ethereal" compared to normal SLOW mode. The best explanation is a brief description of how QUICK works.

Most of you know that, in SLOW mode, the computer is only working on your program during the "margins" above and below the actual TV picture. This happens at a rate of 60 frames (more precisely, "fields") per second. QUICK mode operates by doing different things during alternate fields.

During the odd-numbered fields, the display operates exactly as in SLOW mode. During even-numbered fields, however, the ENTIRE field is used for computing. The net effect is an almost 3x speed increase. The drawback is that the picture is displayed only 30 times per second, interspersed with blank screens. This results in a somewhat ghostly flashing effect that is especially pronounced on inverse graphics. However, the display is at least stable and lacks the jagged edges that result from using PAUSE within your loops in FAST mode.

not for long

what about epilepsy?

might use alternately for games.

HOW IT WORKS

Refer to the disassembly to understand how QUICK works. The routine is invoked with RAND USR 8192, which executes the little routine at 2000h. This first calls SLOW mode, then re-vectors the IX register. The next time the computer encounters a JP (IX) in the interrupt routine, it will be redirected to our display handler instead. We also initialize variable TOGL to contain a pattern of alternating "1"s and "0"s.

The first part of our service routine, from 2012h to 2066h, is a carbon-copy of the main display loop in the ROM. It decrements the FRAMES counter, scans the keyboard and starts the vertical sync pulse, and processes the keyboard debounce information. It then turns off the vertical sync pulse with OUT (FF),A, and sets up HL to point to the first byte of the display file "echo."

We then rotate the TOGL variable. The JP C that follows will cause one of two things to happen, on alternate frames.

If a "0" was rotated out of the TOGL variable, things proceed as usual (FLD1). We generate the top margin, then create the display, then generate the bottom margin. Finally, we jump back to the beginning of the service routine to do it all over.

If a "1" was rotated out of the TOGL variable, then we jump to FLD2. Again, we make the top margin. But then, instead of creating the display, we instead create another "margin," this time 192 (C0h) lines long, in its place. Again we finish off with the bottom margin before returning to the start of the loop. Notice that the computer is working on your program the whole time (except for brief interrupts at the end of each line, for house-keeping purposes).

INSTALLING QUICK MODE

As listed, the program assumes that you have memory in the 8-16K region (as with a 64K pack, or using a SCRAM or similar NVM board). If you don't have memory down there, see "Making Changes," below.

Start by entering a 1 REM line followed by 162 characters. Add the lines of the simple BASIC decimal loader, Listing 1. RUN the listing, and enter the values of Table 1. When done, overwrite Listing 1 with Listing 2 (don't change or delete line 1).

RUN Listing 2 to save to tape. After saving, or later re-loading, the QUICK code will be booted into low memory (starting at 8192). You can now load other software.

USING QUICK MODE

After loading your program, you must invoke QUICK mode with RAND USR 8192, followed by RUN (or GOTO xxx) to start your program. How about that, eh? If you have a hardware video-reverse switch, you'll probably find the pulsations less annoying if you switch to reverse mode.

RAND USR 8204 can be used to exit back to normal SLOW mode. However, entering FAST mode will also wrench control from the QUICK routines. Similarly, PAUSE, LOAD, SAVE, and entering or deleting program lines will also return to the normal display.

You can easily modify your BASIC programs to make use of QUICK mode. Simply replace all FAST commands with RAND USR 8192, and all SLOW commands with RAND USR 8204. You'll find the ghostly display considerably better than staring at a blank screen during FAST mode printing loops, etc., with much less of a speed compromise than if you did everything in SLOW mode.

MAKING CHANGES

The QUICK code can be moved just about anywhere you'd care to put it; e.g. above RAMTOP, in high memory (if you decoded the M1* line), or leave it in a REM statement. Simply change all JPs, and modify the LD IX command accordingly. Similarly, you can choose just about any unused system variable for TOGL; I chose this one because very little existing software uses this location for its variables (Hot Z is a notable exception).

LISTING 1: DECIMAL LOADER

```
  1 REM =======162 SPACES========
 10 REM DECIMAL LOADER
100 FOR A=16514 TO 16675
105 INPUT B
107 POKE A,B
110 LET B$="00"+ STR$ PEEK A
120 PRINT B$( LEN B$-2 TO );":";
130 NEXT A
```

"Quick" Listing 1

use as-is

"Quick" Listing 2 un as-is

TABLE 1: DECIMAL CODE LISTING

```
205:043:015:221:033:018:032:253:
054:122:170:201:221:033:129:002:
201:000:042:052:064:043:062:127:
164:181:124:032:003:023:024:002:
070:055:103:034:052:064:208:205:
187:002:237:075:037:064:034:037:
064:120:198:002:237:066:058:039:
064:180:181:088:006:011:033:059:
064:203:134:032:008:203:126:203:
198:200:005:000:055:033:039:064:
063:203:016:016:254:070:123:254:
254:159:006:031:182:160:031:119:
211:255:042:012:064:203:252:253:
203:122:006:218:130:032:205:146:
002:237:095:001:001:025:062:245:
205:181:002:043:205:146:002:195:
018:032:205:146:002:253:054:040:
192:205:146:002:253:054:040:031:
205:146:002:195:018:032:001:150:
000:033:130:064:017:000:032:237:
176:201:
```

234

1654

LISTING 2

LOAD "QUICKPGM"
Then LLIST
to 2040

(or use this as-is)

```
   1~REM ======162 SPACES=======

  10 SAVE "QUICK"
  20 PRINT "ENABLE 8-10K RAM, TH
EN PRESS    ANY KEY TO INSTALL Q
UICK MODE"
  30 PAUSE 4E4
  40 CLS
  50 PRINT " RAND USR 8192 TO EN
TER QUICK    MODE",,,," USR 8204
 OR FAST TO EXIT"
  60 RAND USR 16664
```

"QUICK" DISASSEMBLY

```
2000~CD2B0F   QUIK CALL SLOW
2003 DD211220      LD IX,DP-Q
2007 FD367AAA      LD (TOGL),AA
200B C9            RET
200C DD218102 NRML LD IX,ZXIX
2010 C9            RET
2011 00            NOP
2012 2A3440   DP-Q LD HL,(FRMS)
2015 2B            DEC HL
2016 3E7F     DP-P LD A,7F
2018 A4            AND H
2019 B5            OR L
201A 7C            LD A,H
201B 2003          JR NZ ANTH
201D 17            RLA
201E 1802          JR O-NC
2020 46       ANTH LD B,(HL)
2021 37            SCF
2022 67       O-NC LD H,A
2023 223440        LD (FRMS),HL
2026 D0            RET NC
2027 CDBB02   DP-2 CALL KBSC
202A ED4B2540      LD BC,(LASK)
202E 222540        LD (LASK),HL
2031 78            LD A,B
2032 C602          ADD A,02
2034 ED42          SBC HL,BC
2036 3A2740        LD A,(DBNC)
2039 B4            OR H
203A B5            OR L
203B 58            LD E,B
203C 060B          LD B,0B
203E 213B40        LD HL,CDFG
2041 CB86          RES 0,(HL)
2043 2008          JR NZ NKEY
2045 CB7E          BIT 7,(HL)
2047 CBC6          SET 0,(HL)
2049 C8            RET Z
204A 05            DEC B
204B 00            NOP
204C 37            SCF
204D 212740   NKEY LD HL,DBNC
2050 3F            CCF
2051 CB10          RL B
```

Disassembly Part 1

use as-is

```
2053~10FE     LP-B DJNZ LP-B
2055 46            LD B,(HL)
2056 7B            LD A,E
2057 FEFE          CP FE
2059 9F            SBC A,A
205A 061F          LD B,1F
205C B6            OR (HL)
205D A0            AND B
205E 1F            RRA
205F 77            LD (HL),A
2060 D3FF          OUT (FF),A
2062 2A0C40        LD HL,(DFIL)
2065 CBFC          SET 7,H
2067 FDCB7A06      RLC (TOGL)
206B DA8220        JP C FLD2
206E CD9202   FLD1 CALL DP-3
2071 ED5F          LD A,R
2073 010119        LD BC,1901
2076 3EF5          LD A,F5
2078 CDB502        CALL DP-5
207B 2B            DEC HL
207C CD9202        CALL DP-3
207F C31220        JP DP-Q
2082 CD9202   FLD2 CALL DP-3
2085 FD3628C0      LD (MARG),C0
2089 CD9202        CALL DP-3
208C FD36281F      LD (MARG),1F
2090 CD9202        CALL DP-3
2093 C31220        JP DP-Q
2096 019600   BOOT LD BC,0096
2099 218240        LD HL,1REM
209C 110020        LD DE,QUIK
209F EDB0          LDIR
20A1 C9            RET
```

Disassembly
Part 2

Use as-is